

1

Pour commencer

OpenCV est une librairie de traitement d'images écrite en C++, open-source, dont les sources sont téléchargeables sur opencv.org. Cette librairie est disponible pour les systèmes Linux, macOS, Windows, Android et iOS. Des interfaces pour les langages Python, Java et Matlab sont proposées. Un portage pour C# existe aussi mais il n'est pas maintenu par le site.

La licence d'OpenCV est [3-Clause BSD License](#), une des licences les moins restrictives. OpenCV peut être intégré dans du code propriétaire. De ce fait, la librairie OpenCV est utilisée dans de nombreux domaines comme la vidéo surveillance, la vidéo sur le web, les drones, deep learning et utilisée par de nombreuses sociétés comme IBM, Microsoft, Intel, Google, etc.

La librairie OpenCV est, en réalité, divisée en deux librairies `opencv` et `opencv_contrib`, organisées en modules thématiques. Au niveau du C++ `opencv_contrib` a besoin du dépôt `opencv` pour être utilisable. Pour les paquets Python ce n'est pas le cas, il faut soit utiliser le paquet `opencv` soit le paquet `opencv_contrib` mais en aucun cas les deux. Le paquet `opencv_contrib` contient `opencv` et quelques algorithmes en cours de test. La différence de taille entre les paquets `opencv` et `opencv_contrib` est d'un peu moins 10 Mo.

Pour installer OpenCV, il faut utiliser le gestionnaire de paquets.

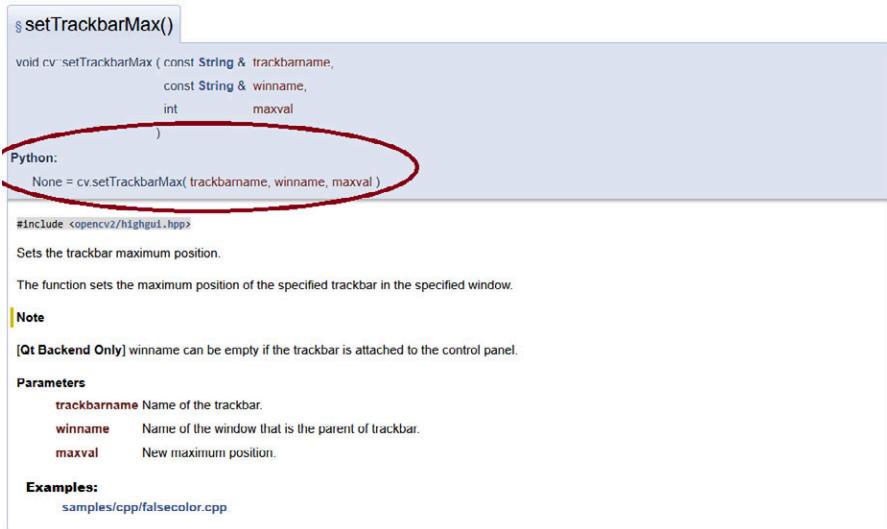
Pour l'écriture de ce livre, nous avons dû corriger quelques bugs ou oubli (`cv.copyTo` avec un masque, `cv.ml.KNearest_load` pour kNearest, `facemark fit` et l'architecture du modèle `facedetector`) et ajouter des fonctions pour accéder à plusieurs fonctionnalités du module Stitching. Il est donc recommandé d'utiliser la dernière version d'`opencv_contrib`, soit une version supérieure à 4.1.2.

Version d'OpenCV

La version d'OpenCV utilisée dans ce livre est la version 4.1.2 d'octobre 2019.

La documentation d'OpenCV est accessible sur le [site officiel](#). Il n'y a pas de documentation propre à Python, il faut aller dans la documentation C++ pour voir si un lien sur la fonction Python existe.

Figure 1.1 : Documentation Python disponible depuis la documentation C++



setTrackbarMax()

```
void cv::setTrackbarMax ( const String & trackbarname,
                        const String & winname,
                        int maxval
                      )
```

Python:

```
None = cv.setTrackbarMax( trackbarname, winname, maxval )
```

#include <opencv2/highgui.hpp>

Sets the trackbar maximum position.

The function sets the maximum position of the specified trackbar in the specified window.

Note

[Qt Backend Only] winname can be empty if the trackbar is attached to the control panel.

Parameters

- trackbarname Name of the trackbar.
- winname Name of the window that is the parent of trackbar.
- maxval New maximum position.

Examples:

[samples/cpp/falsecolor.cpp](#)

Les tutoriels Python sont accessibles sur une [page dédiée](#). L'ensemble des exemples Python sont dans le [dossier python du dépôt GitHub d'OpenCV](#).

1.1. Organisation en modules d'OpenCV

Les modules d'OpenCV sont les suivants :

- [core](#) pour la gestion de la mémoire et la gestion de librairie spécifique (OpenGL, DirectX, etc.) ;
- [imgproc](#) pour le filtrage des images, les transformations géométriques, histogrammes, seuillage, etc. ;
- [imgcodecs](#) pour la lecture et l'écriture d'images ;
- [highgui](#) pour le fenêtrage ;
- [videoio](#) pour la capture d'images et la lecture ou l'écriture de vidéo ;
- [video](#) pour l'analyse de mouvement ou le suivi de trajectoire ;

- `calib3d` pour l'étalonnage (calibrage) de caméra ou de système de stéréovision et les algorithmes de stéréovision ;
- `features2d` pour la détection et le calcul des attributs ;
- `objdetect` pour la détection d'objet à l'aide de détecteur en cascade ou d'histogramme de gradient orienté ;
- `dnn` pour l'utilisation des modèles entraînés avec les librairies Caffe, Torch ou TensorFlow ;
- `ml` pour l'entraînement et l'utilisation des modèles d'apprentissage ;
- `flann` pour le partitionnement de données (utilisation de la librairie Fast Library for Approximate Nearest Neighbors) ;
- `photo` pour l'amélioration d'images ;
- `stitching` pour la création de photographie panoramique.

`opencv_contrib` est une librairie complémentaire, contenant des implémentations d'algorithmes en cours de validation proposées par des contributeurs ainsi que la librairie principale dans le cas de Python. Ces implémentations, regroupées en `modules`, sont testées par les utilisateurs, et une fois celles-ci validées, elles sont insérées dans la librairie principale opencv. Par exemple, le module `dnn` était dans `opencv_contrib` dans la version 3.2 et a été inséré depuis dans OpenCV version 3.3.

Dans ce livre, nous utiliserons les modules suivants d'`opencv_contrib` :

- `aruco` pour l'utilisation des marqueurs de type ArUco ;
- `face` pour la reconnaissance faciale ;
- `ximgproc` contient quelques algorithmes de filtrages et un algorithme de squelettisation ;
- `xphoto` contient en particulier l'algorithme `oilPainting`.

1.2. Installation d'OpenCV

Pour installer `opencv_contrib`, il faut d'abord installer `numpy` puis `opencv_contrib` :

```
pip install numpy
pip install opencv-contrib-python
```

On peut vérifier que l'installation est correcte en entrant les commandes suivantes :

```
import numpy as np
import cv2 as cv
print(cv.getBuildInformation())
```

Le résultat contient la version d'opencv_contrib et toutes les librairies utilisées pour construire cette version :

```
General configuration for OpenCV 4.1.1-dev
=====
Version control:               4.0.1-1212-g3debb9880
Extra modules:
  Location (extra):          G:/Lib/opencv_contrib/modules
  Version control (extra):    3.4.1-567-g0915b7ea
```

1.3. Que trouve-t-on dans ce livre ?

Le chapitre [Écran de contrôle de caméra vidéo – Gestion des processus légers](#) ainsi que l'annexe [Memento d'OpenCV](#) permettent aux lecteurs de se familiariser avec les bases d'OpenCV 4 en Python. Les autres chapitres sont relativement indépendants et exposent les principales fonctions disponibles en OpenCV.

Matériel nécessaire

Pour les chapitres [Écran de contrôle de caméra vidéo – Gestion des processus légers](#) et [Réaliser une caméra panoramique](#), il faut avoir au moins deux caméras. Dans ce livre, des webcams Logitech C270 ont été utilisées.

Le chapitre [Segmentation d'images](#) présente les fonctions de segmentation d'images comme la détection de contours, le seuillage ou l'utilisation de l'algorithme k-moyenne.

Le chapitre [Réaliser une caméra panoramique](#) permet de réaliser une image panoramique à partir de plusieurs caméras en utilisant le module `Stitching`.

Le chapitre [Calibrage d'images et stéréovision](#) permet la réalisation complète d'un système de stéréovision composée de deux caméras : calibrage, appariement des images, mesures des coordonnées et visualisation en trois dimensions. Pour ce chapitre, il est nécessaire d'avoir VTK installé.

Le chapitre [Apprentissage automatique – Machine Learning](#) permet de voir les différents types d'apprentissages automatiques (*Machine Learning*) disponibles dans OpenCV.

Le chapitre [Utilisation des modèles de deep learning](#) présente les différentes fonctions permettant d'utiliser un modèle pré-entraîné de deep learning.

Le chapitre [Traitement non réaliste de photographies](#) montre l'utilisation de quelques fonctions disponibles dans les modules photo ou xphoto. Un exemple d'utilisation de deep learning (Neural Algorithm of Artistic Style) est présenté.

Enfin le dernier chapitre [Reconnaissance faciale et de la pose](#) montre comment effectuer une reconnaissance faciale en utilisant le module face.

En annexe, vous trouverez des compléments d'information sur la [construction d'OpenCV à partir des sources](#), son [installation sur Raspberry](#) et un rappel de quelques [notions de base](#) pour utiliser OpenCV avec le langage Python.

1.4. Exemples du livre

Ce livre contient une vingtaine d'exemples. Ils sont disponibles sur le [dépôt GitHub Traitement_d-images_et_de_videos_avec_OpenCV4_en_Python](#). Au début de chaque exemple, nous vous indiquons dans quel dossier se trouve son code source.

Par ailleurs, vous pouvez afficher séparément, sur une tablette ou un smartphone, un grand nombre de listings avec leurs références numérotées pour les visualiser en regard des explications. Il suffit pour cela de cliquer sur le picto  qui les précède. La partie du listing concernée s'affiche alors dans votre navigateur web.

Les exemples présentés dans ce livre s'appuient sur différents travaux de recherche ; vous trouverez la liste des publications afférantes à l'annexe [Bibliographie](#). Elles sont référencées au fil des chapitres sous la forme [Abc] où Abc correspond aux premières lettres du nom de l'auteur principal.

URL raccourcies

Dans un souci de lisibilité, et pour pouvoir les maintenir à jour, nous avons pris le parti de remplacer toutes les adresses internet par ce qu'on appelle des URL raccourcies. Une fois que vous avez accédé à la page cible, nous vous invitons à l'enregistrer avec un marque-page si vous souhaitez y revenir fréquemment. Vous disposerez alors du lien direct. Si celui-ci se périmé, n'hésitez pas à repasser par l'URL raccourcie. Si cette dernière aussi échoue, vous pouvez nous le signaler !